

1概述

2019年3月11日 10:18



◆ 计算机系统简介

1- 软硬件

1. 硬件：实体部分，看得见摸得着的电子元器件
2. 软件：看不见摸不着的程序
 - (1) 系统软件：管理系统，调度资源
 - (2) 应用软件：为满足用户任务编制的

2- 层次结构

- 第三级：虚拟机器M4（高级语言机器）用编译程序翻译成汇编语言程序
- 第二级：虚拟机器M3（汇编语言机器）用汇编语言翻译成机器语言程序
- 操作系统：虚拟机器M2（操作系统机器）用机器语言解释操作系统
- 第一级：传统机器M1（传统机器M1）用微程序解释机器命令
- 第零级：微程序机器M0（微指令系统）由硬件直接执行微指令

3- 组成和体系结构

1. 结构architecture：指令系统等能体现机器属性的结构、功能特性
2. 组成organization：结构系统的实现方法
3. 实现：组成的物理实现

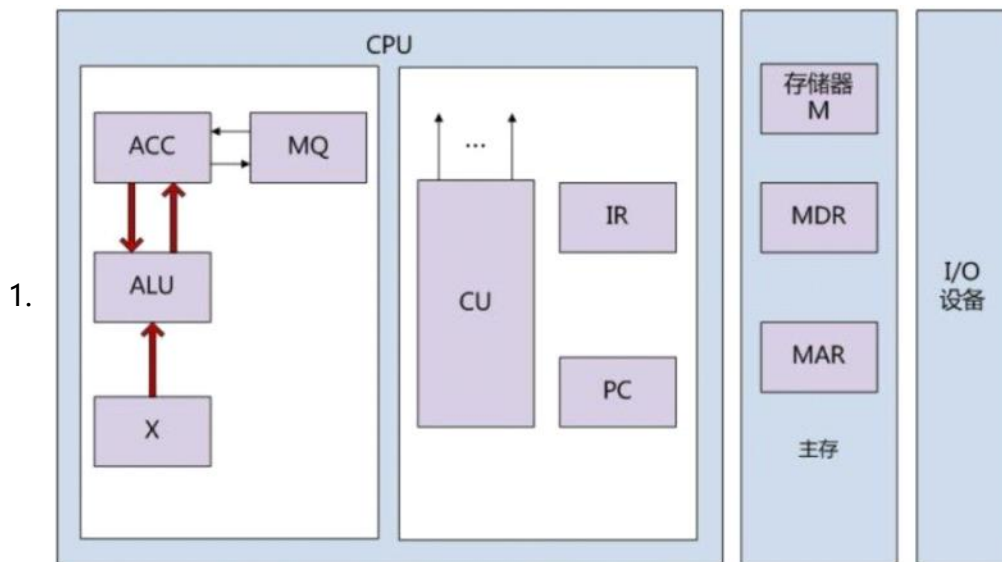


◆ 计算机的基本组成

1- von Neumann机

1. 由**运算器、存储器、控制器、输入设备和输出设备**五大部件组成
2. 指令(程序)和数据以**二进制**不加区别地存储在存储器中、可按址寻访
3. 指令由操作码和地址码组成，操作码告诉机器操作的性质, 地址码说明要在哪里操作
4. 指令一般顺序存放，顺序执行，偶尔根据条件改变顺序
5. 以运算器为核心, 输出输入设备与存储器之间的数据传输通过运算器来完成

2- 硬件框图



2. CPU(Central Processing Unit) 中央处理器

- (1) 核心部件为ALU(Arithmetic Logic Unit,算术逻辑单元)
- (2) 和CU(Control Unit,控制单元)

3. I/O设备，受CU控制，有时称为外部设备

4. MM(Main Memory)主存储器，分为RAM随机存储器和ROM只读存储器

- (1) CPU与MM合成主机，I/O设备可称为外部设备

3- 工作步骤

(1) Ad(Address) 形式地址

(2) DR(Data Register) 数据寄存器

(3) AR(Address Register) 地址寄存器(MAR)

(4) IR(Instruction Register) 指令寄存器

(5) BR(Buffer Register) 缓冲寄存器(MBR)

(6) ID(Instruction Decoder) 指令译码器

(7) PC(ProgramCounter) 程序计数器

(8) ACC(Accumulator) 累加器

(9) MQ(Multiplier-Quotient Register) 乘商寄存器

1. 取指令阶段：将一条指令从主存中取到指令寄存器

- (1) 程序计数器PC中的数值，用来指示当前指令在主存中的位置。当一条指令被取出后，PC中的数值将根据指令字长度而自动递增
- (2) //PC -> AR -> Memory
- (3) //Memory -> IR

2. 指令译码阶段 (Instruction Decode, ID)：指令译码器ID按预定的格式，对取回的指令拆分和解释，识别出不同的指令类别及获取操作数的方法

- (1) 在组合逻辑控制的计算机中，指令译码器对不同的指令操作码产生不同的控制电位，以形成不同的微操作序列
- (2) 在微程序控制的计算机中，指令译码器用指令操作码来找到执行该指令的微程序的入口，并从此入口开始执行
- (3) //
- (4) //Memory -> IR -> ID -> { 2.PC变化

(5) // { 3.CU (Control Unit)

3. 访存取数阶段 (Memory, MEM) : 根据指令地址码, 得到操作数在主存中的地址, 并从主存中读取该操作数用于运算

(1) //Ad -> AR -> AD -> Memory

4. 执行指令阶段 (Execute, EX) : 完成指令所规定的各种操作, 具体实现指令的功能。为此, CPU的不同部分被连接起来, 以执行所需的操作

(1) 如, 加法运算, 算术逻辑单元ALU将被连接到一组输入和一组输出, 输入端提供需要相加的数值, 输出端将含有最后的运算结果。

(2) //Memory -> DR -> ALU

5. 结果写回阶段 (Writeback, WB) : 把执行指令阶段的运行结果数据 “写回” 到某种存储形式

(1) 结果数据经常被写到CPU的内部寄存器中, 以便被后续的指令快速地存取; 在有些情况下, 结果数据也可被写入相对较慢、但较廉价且容量较大的主存。许多指令还会改变程序状态字寄存器中标志位的状态, 这些标志位标识着不同的操作结果, 可被用来影响程序的动作

(2) //DR -> Memory

6. 循环阶段: 在指令执行完毕、结果数据写回之后, 若无意外事件 (如结果溢出等) 发生, 计算机就接着从程序计数器PC中取得下一条指令地址, 开始新一轮的循环, 下一个指令周期将顺序取出下一条指令

(1) //重复 1~5

(2) //遇hlt(holt on)停止



◆ 硬件的主要技术指标

1- 机器字长: CPU一次能处理的位数

2- 存储容量=存储单元数x存储子长

3- 运算速度

1. Gibson法: 各条指令的执行时间乘以指令所占的百分比的加权和

(1) 即 $T_m = \sum f_i \cdot t_i$, 其中 T_m 为运行速度, f_i 为 i 指令占比, t_i 为 i 指令执行时间

2. MIPS法: Million Instruction Per Second, 每秒执行几百万条指令

(1) 如每秒能执行两百万指令的记作2MIPS

(2) 类似的有针对小数运算效率的FLOPS法: Floating Point Operation Per Second

3. CPI法: Cycle Per Instruction, 执行一条指令所需的时钟周期数

(1) 单位是时钟周期 T_c , 即机器主频 R_c 的倒数

4- 加速比例题

1. 如果部件1和部件2的可改进比例都为30%, 那么当部件3的可改进比例为多少时, 系统的加速比才可以达到10%?

(1) 由多部件可改进情况下, Amdahl定律的扩展: $S_n = 1 / ((1 - \sum F_i) + \sum (F_i / S_i))$

知: $10 = 1 / ((1 - 0.3 - 0.3 - F_3) + (0.3 / 30 + 0.3 / 20 + F_3 / 10))$ 得到: $F_3 = 0.36$

故第三部件可改进比为36%

2. 如果3个部件的可改进比例分别为30%、30%、20%, 3个部件同时改进, 那么系统中不可改进部分的执行时间在总执行时间中占的比例是多少?

(1) 设系统改进前执行时间为 T , 则3个部件的执行时间是 $(0.3 + 0.3 + 0.2)T = 0.8T$, 则不改进部分执行时间为

0.2T。

已知3个部件改进后的速度比分别为：30、20、10，因此3个部件改进后的执行时间为： $T' = 0.3T/30 + 0.3T/20 + 0.2T/10 = 0.045T$

则改进后整个系统的执行时间为 $T_n = 0.045T + 0.2T = 0.245T$ 。

则系统不可改进部分的执行时间占总执行时间的比例为： $0.2T/0.245T = 82\%$,即占到82%

(2)

(3)

(4)

(5)

(6)

(7)

(8) -----我是底线-----